DIGITAL COMPUTER PLATFORMS LAB MANUAL

Department of Electronics & Communication Engineering

VEMU INSTITUTE OF TECHNOLOGY::P.KOTHAKOTA

NEAR PAKALA, CHITTOOR-517112

(Approved by AICTE, New Delhi & Affiliated to JNTUA, Anantapuramu)

DIGITAL COMPUTER PLATFORMS LAB MANUAL

Name:____

H.T.No:____

Year/Semester:______________________________________

Department of Electronics & Communication Engineering

VEMU INSTITUTE OF TECHNOLOGY::P.KOTHAKOTA
NEAR PAKALA, CHITTOOR-517112
(Approved by AICTE, New Delhi & Affiliated to JNTUA, Anantapuramu)

VEMU Institute of Technology
Dept. of Electrical and Electronics Engineering

Vision of the institute

To be one of the premier institutes for professional education producing dynamic and vibrant force of technocrats with competent skills, innovative ideas and leadership qualities to serve the society with ethical and benevolent approach.

Mission of the institute

Mission_1: To create a learning environment with state-of-the art infrastructure, well equipped laboratories, research facilities and qualified senior faculty to impart high quality technical education.

Mission_2: To facilitate the learners to inculcate competent research skills and innovative ideas by Industry-Institute Interaction.

Mission_3: To develop hard work, honesty, leadership qualities and sense of direction in learners by providing value based education.

Vision of the department

To develop as a center of excellence in the Electronics and Communication Engineering field and produce graduates with Technical Skills, Competency, Quality, and Professional Ethics to meet the challenges of the Industry and evolving Society.

Mission of the department

Mission_1: To enrich Technical Skills of students through Effective Teaching and Learning practices to exchange ideas and dissemination of knowledge.

Mission_2: To enable students to develop skill sets through adequate facilities, training on core and multidisciplinary technologies and Competency Enhancement Programs.

Mission_3: To provide training, instill creative thinking and research attitude to the students through Industry-Institute Interaction along with Professional Ethics and values.

Programme Educational Objectives (PEOs)

PEO 1: To prepare the graduates to be able to plan, analyze and provide innovative ideas to investigate complex engineering problems of industry in the field of Electronics and Communication Engineering using contemporary design and simulation tools.

PEO-2: To provide students with solid fundamentals in core and multidisciplinary domain for successful implementation of engineering products and also to pursue higher studies.

PEO-3: To inculcate learners with professional and ethical attitude, effective communication skills, teamwork skills, and an ability to relate engineering issues to broader social context at work place

Programme Outcomes(Pos)

PO_1	**Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO_2	**Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO_3	**Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO_4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO_5	**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO_6	**The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO_7	**Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO_8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO_9	**Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO_10	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO_11	**Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO_12	**Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Programme Specific Outcome(PSOs)

PSO_1	**Higher Education :** Qualify in competitive examination for pursuing higher education by applying the fundamental concepts of Electronics and Communication Engineering domains such as Analog & Digital Electronics, Signal Processing, Communication & Networking, Embeded Systems, VLSI Design and Control systems etc.,
PSO_2	**Employment**: Get employed in allied industries through their proficiency in program specific domain knowledge, Specalized software packages and Computer programming or became an entrepreneur.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

III B.Tech. II-Sem (EEE)

(19A02601P) DIGITAL COMPUTE PLATFORMS LAB MANUAL

COURSE OUTCOMES(CO$_{S)}$

CO	Description	BL
CO1	Write and Execute different programs 8086, 8051 & ARM Cortex M0 assembly level languages using MASAM assembler.	02
CO2	Design and implement some specific real time applications	04

PART A:

LIST OF PROGRAMS USING MASAM/ALP:

1. Programs for 16 bit arithmetic operations for 8086 (using various addressing modes) .
2. Program for sorting an array for 8086.
3. Program for searching for a number or character in a string for 8086.
4. Program for String manipulations for 8086 .

PART-B: List of eexperiments using 8086 and 8051 modules:

1. Interfacing ADC and DAC to 8086.
2. Parallel communication between two microprocessors using 8255.
3. Serial communication between two microprocessor kits using 8251.
4. Interfacing to 8086 and programming to control stepper motor.
5. Programming using arithmetic, logical and bit manipulation instructions of 8051
6. Program and verify Timer/Counter in 8051.
7. Program and verify interrupt handling in 8051.
8. UART operation in 8051.
9. Communication between 8051 kit and PC.
10. Interfacing LCD to 8051.
11. Interfacing matrix or keyboard to 8051.

VEMU INSTITUTE OF TECHNOLOGY::P.KOTHAKOTA

NEAR PAKALA, CHITTOOR-517112

(Approved by AICTE, New Delhi & Affiliated to JNTUA, Anantapuramu)

Dept. of Electrical and Electronics Engineering

(19A02601P) DIGITAL LOGIC DESIGN

III B.Tech-II SEM

LIST OF EXPERIMENTS TO BE CONDUCTED

PART A:

LIST OF PROGRAMS USING MASAM/ALP:

1.Programs for 16 bit arithmetic operations for 8086 (using various addressing modes) .

2.Program for sorting an array for 8086.

3.Program for searching for a number or character in a string for 8086.

4.Program for String manipulations for 8086 .

PART-B: List of experiments using 8086 and 8051 modules:

1.Interfacing ADC and DAC to 8086.

2.Parallel communication between two microprocessors using 8255.

3.Interfacing to 8086 and programming to control stepper motor.

4.Programming using arithmetic, logical and bit manipulation instructions of 8051

5.Program and verify Timer/Counter in 8051.

6.Program and verify interrupt handling in 8051.

7.UART operation in 8051.

8.Interfacing matrix or keyboard to 8051.

ADDITIONAL EXPERIMENTS (BEYOND SYALLBUS)

1. Interfacing with 8086 -8255 programmable peripheral interface

2.Interfacing with 8051-8279 keyboard interface

CONTENTS

DOS & DONTS IN LABORATORY

DO’s

1. Students should be punctual and regular to the laboratory.
2. Students should come to the lab in-time with proper dress code.
3. Students should maintain discipline all the time and obey the instructions.
4. Students should carry observation and record completed in all aspects.
5. Students should be at their concerned experiment table, unnecessary moment is restricted.
6. Students should follow the indent procedure to receive and deposit the components from lab technician.
7. While doing the experiments any failure/malfunction must be reported to the faculty.
8. Students should check the connections of circuit properly before switch ON the power supply.
9. Students should verify the reading with the help of the lab instructor after completion of experiment.
10. Students must endure that all switches are in the lab OFF position, all the connections are removed.
11. At the end of practical class the apparatus should be returned to the lab technician and take back the indent slip.
12. After completing your lab session SHUTDOWN the systems, TURNOFF the power switches and arrange the chairs properly.
13. Each experiment should be written in the record note book only after getting signature from the lab in charge in the observation notebook.

DON’Ts

1. Don’t eat and drink in the laboratory.
2. Don’t touch electric wires.
3. Don’t turn ON the circuit unless it is completed.
4. Avoid making loose connections.
5. Don’t leave the lab without permission.
6. Don’t bring mobiles into laboratory.
7. Do not open any irrelevant sites on computer.
8. Don’t use a flash drive on computers.

SCHEME OF EVALUATION

S.No	Program	Date	Marks Awarded				Total 30(M)
			Record (10M)	Obs. (10M)	Viva (5M)	Attd. (5M)	
PART-A							
1	Programs for 16 bit arithmetic operations for 8086 (using various addressing modes)						
2	Program for sorting an array for 8086.						
3	Program for searching for a number or character in a string for 8086.						
4	Program for String manipulations for 8086 .						
PART-B							
5	Interfacing ADC and DAC to 8086						
6	Parallel communication between two microprocessors using 8255.						
7	Interfacing to 8086 and programming to control stepper motor						
8	Programming using arithmetic, logical and bit manipulation instructions of 8051						
9	Program and verify Timer/Counter in 8051.						
10	Program and verify interrupt handling in 8051.						
11	UART operation in 8051.						
12	Interfacing matrix or keyboard to 8051						
ADVANCED EXPERIMENTS (Beyond the Curriculum)							
13	Interfacing with 8086 -8255 programmable peripheral interface						
14	Interfacing with 8051-8279 keyboard interface						

Signature of Lab In-charge

PART-I

EXP NO.1 **DATE:**

PROGRAMS FOR 16 BIT ARITHMETIC OPERATIONS FOR8086 (USING VARIOUS ADDRESSING MODES).

A) ADDITION:

i) 16 BIT ADDITION (DIRECT ADDERESSING MODE):

AIM: - To write an assembly language program for Addition of two 16-bit numbers.

APPARATUS: 1. 8086 microprocessor kit/MASM --- 1
2. RPS (+5V) ---- 1

PROGRAM:

```
ASSUME CS: CODE, DS :DATA
DATA SEGMENT
NUM DW 1232H
DATA ENDS
CODE SEGMENT
START:
            MOV AX ,DATA
            MOV DS, AX
            MOV AX,4562H
            ADD AX,NUM
            INT 03H
CODE ENDS
END START
```

OPCODE:

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

ii) 16 BIT ADDITION(IMMEDIATE ADDERESSING MODE):

AIM: - To write an assembly language program for Addition of two 16-bit numbers.

APPARATUS: 1. 8086 microprocessor kit/MASM --- 1
2. RPS (+5V) 1

PROGRAM:

```
  ASSUME CS:CODE
CODE SEGMENT
START:
        MOV     BX,5678H
        ADD     AX,1234H
        INT 03H
CODE    ENDS
END START
```

OPCODE:

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

iii) 16 BIT ADDITION(INDIRECTADDERESSING MODE):

AIM: - To write an assembly language program for Addition of two 16-bit numbers.

APPARATUS: 1. 8086 microprocessor kit/MASM --- 1
2. RPS (+5V)----1

PROGRAM:

```
ASSUME CS:CODE,DS:DATADATA
SEGMENT
NUM DW 02H
DATA ENDS
CODE SEGMENT
START:
        MOV AX,DATA
        MOV DS,AX
        MOV AX,4444H
        MOV BX,OFFSET NUM
        ADD AX,[BX]
        INT 03H
        CODE ENDS
        END START
```

OPCODE:

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

iv) 16 BIT ADDITION (REGISTERADDERESSING MODE):

AIM: - To write an assembly language program for Addition of two 16-bit numbers.

APPARATUS: 1. 8086 microprocessor kit/MASM --- 1
2. RPS (+5V) ---- 1

PROGRAM:

```
ASSUME CS:CODE

CODE SEGMENT

START:
          MOV BX,5678H
          MOV AX,1234H
           ADD AX,BX
          INT 03H
CODE ENDS
END START
```

OPCODE:

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

B) SUBTRACTION:

i) 16 BIT SUBTRACTION (DIRECT ADDRESSING MODE):

AIM: - To write an assembly language program for subtraction of two 16-bitnumbers.

APPARATUS: 1. 8086 microprocessor kit/MASM ----------- 1
2. RPS (+5V) ------- 1

PROGRAM:

```
ASSUME CS:CODE,DS:DATA

DATA  SEGMENT
NUM DW 1232H
DATA ENDS
 CODE  SEGMENT
START:
        MOV AX,DATA
        MOV DS,AX
        MOV AX,4562H
        SUB AX,NUM
        INT 03H
CODE ENDS
END START
```

OPCODE:

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

ii) 16 BIT SUBTRACTION (IMMEDIATE ADDRESSING MODE):

AIM: - To write an assembly language program for subtraction of two 16-bit numbers.

APPARATUS: 1. 8086 microprocessor kit/MASM --- 1
2. RPS (+5V) ---- 1

PROGRAM:

```
ASSUME CS:CODE

CODE SEGMENT

START:

        MOV AX,5678H

        SUB AX,1234H

        INT 03H

CODE ENDS

END START
```

OPCODE:

MEMORY LOCATION	**OP-CODE**	**LABEL**	**INSTRUCTION**

INPUT:

OUTPUT:

iii) 16 BIT SUBTRACTION (INDIRECTADDRESSING MODE):

AIM: - To write an assembly language program for subtraction of two 16-bit numbers.

APPARATUS: 1. 8086 microprocessor kit/MASM --- 1
2. RPS (+5V)----1

PROGRAM:

```
ASSUME CS:CODE,DS:DATA

DATA SEGMENT

NUM DW 02H

DATA ENDS

CODE SEGMENT

START:

          MOV AX,DATA

          MOV DS,AX

          MOV AX,4444H

          MOV BX,OFFSET NUM

          SUB AX,[BX]

          INT 03H

CODE ENDS

END START
```

OPCODE:

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

iv) 16 BIT SUBTRACTION (REGISTERADDRESSING MODE):

AIM: - To write an assembly language program for subtraction of two 16-bit numbers.

APPARATUS: 1. 8086 microprocessor kit/MASM --- 1
2. RPS (+5V)----1

PROGRAM:

```
ASSUME CS:CODE
CODE SEGMENT
START:
        MOV BX,5678H
        MOV AX,1234H
        SUB AX,BX
        INT 03H
CODE ENDS
END START
```

OPCODE:

MEMORY LOCATION	**OP-CODE**	**LABEL**	**INSTRUCTION**

INPUT:

OUTPUT:

C) MULTIPLICATION:

i) 16 BIT MULTIPLICATION (DIRECT ADDERESSING MODE):

AIM: - To write an assembly language program for multiplication of two 16-bit numbers.

APPARATUS: 1. 8086 microprocessor kit/MASM --- 1
2. RPS (+5V)---- 1

PROGRAM:

```
ASSUME CS:CODE,DS:DATA
DATA SEGMENT
X DB 02H
DATA ENDS
CODE SEGMENT
START:
        MOV AX,DATA
        MOV DS,AX
        MOV AX,0002H
        MUL X
        INT 03H

CODE ENDS
END START
```

OPCODE:

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

C) MULTIPLICATION:

ii) 16 BIT MULTIPLICATION (REGISTER ADDERESSING MODE):

AIM: - To write an assembly language program for multiplication of two 16-bit numbers.

APPARATUS: 1. 8086 microprocessor kit/MASM --- 1
2. RPS (+5V)---- 1

PROGRAM:

```
ASSUME CS:CODE
CODE SEGMENT
START:
        MOV BX,1234H
        MOV AX,1234H
        MUL BX
        INT 03H
CODE ENDS
END START
```

OPCODE:

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

D) DIVISION:

i) 16 BIT DIVISION (DIRECT ADDRESSING MODE):

AIM: - To write an assembly language program for multiplication of two 16-bit numbers.

APPARATUS: 1. 8086 microprocessor kit/MASM --- 1
2. RPS (+5V)----1

PROGRAM:

```
ASSUME CS:CODE,DS:DATA
DATA SEGMENT
NUM DW 02H
DATA ENDS
CODE SEGMENT
START:
        MOV AX,DATA
        MOV DS,AX
        MOV AX,4444H
        DIV NUM
        INT 03H
CODE ENDS
END START
```

OPCODE:

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

D) DIVISION:

ii) 16 BIT DIVISION (REGISTER ADDRESSING MODE):

AIM: - To write an assembly language program for multiplication of two 16-bit numbers.

APPARATUS: 1. 8086 microprocessor kit/MASM --- 1
2. RPS (+5V)---- 1

PROGRAM:

```
ASSUME CS:CODE
CODE SEGMENT
START:
        MOV BX,0022H
        MOV AX,4444H
        DIV BX
        INT 03H
CODE ENDS
END START
```

OPCODE:

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

RESULT:

Viva Question:

1) What is meant by microprocessor?

2) What is meant by accumulator?

3) What is meant by assembler directive?

4) What are segment Registers?

5) What is the use of INT03Hinstruction?

EXP NO.:2 **DATE:**

PROGRAM FOR SORTING AN ARRAY FOR 8086

A) ASCENDING ORDER

AIM:-Program to sort the given numbers in ascending order

APPARATUS: 1. 8086 microprocessor kit/MASM --- 1
2. RPS (+5V) ---- 1

PROGRAM:

```
ASSUME CS:CODE,DS:DATA
DATA SEGMENT
LIST DW 55H,67H,23H,45H
COUNT EQU 04H
DATA ENDS
CODE SEGMENT
START:
        MOV AX,DATA
        MOV DS,AX
        MOV DX,COUNT-1
        L1:MOV CX,DX
        MOV SI,OFFSET LIST
        L2:MOV AX,[SI]
        CMP AX,[SI+2]
        JL L3
        XCHG [SI+2],AX
        XCHG [SI],AX
        L3:ADD SI,02
        LOOP L2
        DEC DX
        JNZ L1
        INT 03H
    CODE ENDS
    END START
```

OPCODE:

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

B) DECENDING ORDER

AIM:-Program to sort the given numbers in decending order

APPARATUS: 1. 8086 microprocessor kit/MASM --- 1
2. RPS (+5V)----1

PROGRAM:

```
ASSUME CS:CODE,DS:DATA

DATA SEGMENT

LIST DW 55H,67H,23H,45H

COUNT EQU 04H

DATA ENDS

CODE SEGMENT

START:

        MOV AX,DATA

        MOV DS,AX

        MOV DX,COUNT-1

        L1:MOV CX,DX

        MOV SI,OFFSET LIST

        L2:MOV AX,[SI]

        CMP AX,[SI+2]

        JA L3

        XCHG [SI+2],AX

        XCHG [SI],AX

        L3:ADD SI,02
        LOOP L2
        DEC DX
        JNZ L1
        INT 03H
CODE ENDS
END START
```

OPCODE:

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

RESULT:

Viva Questions:

1) What is the use of SI Register?

2) What is the use of XCHG instruction?

3) What is the use of CX Register ?

4) What is the use of JNZ instruction?

5)State the logic behind the Sorting an array of data in Descending order

EXP NO: 3 **DATE:**

PROGRAM FOR SEARCHING FOR A NUMBER ORCHARACTER IN A STRING FOR 8086.

A) SERCHING OF AN ARRAY

AIM: Write an alp program for to search a number or character from an array.

APPARATUS: 1. 8086 microprocessor kit/MASM --- 1
2. RPS (+5V) ---- 1

PROGRAM:

SEARCHING AN ARRAY CASE: EQUAL

```
ASSUME CS: CODE, DS: DATA
DATA SEGMENT
ARRAY DB 27H, 0A9H, 82H, 4DH, 36HN1 DB
82H
N2 DB 25H COUNT
DB 05H DATA
ENDS CODE
SEGMENTSTART:
        MOV AX, DATA
        MOV DS, AX MOV
        CL, COUNT
        MOV BX, OFFSET ARRAYLEA
        BX, ARRAY
        MOV DL, N1
 BACK: CMP DL,[BX]E
        EXIT
        INC BX LOOP
        BACK
        MOV AX, 0FFFFH
        INT 03H
 EXIT: MOV AX, 00H
        INT 03H
CODE ENDS
END START
```

SEARCHING AN ARRAY CASE: NOT EQUAL

```
ASSUME CS: CODE, DS: DATA
DATA SEGMENT
ARRAY DB 27H, 0A9H, 82H, 4DH, 36H
N1 DB 82H
N2 DB 25H
COUNT DB 05H
DATA ENDS
CODE SEGMENT
START:
        MOV AX, DATA
        MOV DS, AX MOV
        CL, COUNT
        MOV BX, OFFSET ARRAYLEA
        BX, ARRAY
        MOV DL, N2

        BACK: CMP DL, [BX]JE
        EXIT
        INC BX LOOP
        BACK
        MOV AX, 0FFFFH
        INT 03H
        EXIT: MOV AX, 00H
        INT 03H
CODE ENDS
END START
```

OPCODE:(CASE:EQUAL)

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

OPCODE(CASE:NOT EQUAL)

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

B)SERCHING A CHARACTER

AIM: Write an alp program for to search a number or character from a string.

APPARATUS: 1. 8086 microprocessor kit/MASM --- 1
2. RPS (+5V) ---- 1

PROGRAM:

```
ASSUME CS:CODE,ES:DATA

DATA SEGMENT

STG DB "ANURAG COLLEGE OF ENGINEERING"CHAR1 DB "C"

CHAR2 DB "Z"

M1 DB "CHARACTER FOUND $"

M2 DB "CHARACTER NOT FOUND$"DATA

ENDS

CODE SEGMENT

START:

        MOV AX,DATA

        MOV ES,AX STD

        LEA DI,[STG+0AH]

        MOV AL,CHAR1
```

```
    MOV CX,28
    REPNZ SCASBJZ
    FOUND LEA
    DX,M2 JMP XYZ
    FOUND:LEA DX,M1
    XYZ:MOV AH,09H INT
    21H
    MOV AH,4CH
    INT 21H
CODE ENDS
END START
```

OPCODE:

MEMORY LOCATION	**OP-CODE**	**LABEL**	**INSTRUCTION**

INPUT:

OUTPUT:

RESULT:

Viva Questions:

1) What is the use of SCASB Register?

2) What is the use of REPNE instruction?

3) What is the relation of CX Register with REPNE?

4) Which interrupts are generally used for critical events?

5) Which Stack is used in 8086? 3. What is SIM and RIM instructions

EXP NO.:4 **DATE:**

PROGRAM FOR STRING MANIPULATIONS FOR 8086.

A) MOVING BLOCK OF DATA FROM ONE MEMORY LOCATION TOANOTHER MEMORY LOCATION

AIM: To write an alp for transfer block of data from one memory location to another memory location.

APPARATUS: 1. 8086 microprocessor kit/MASM --- 1
2. RPS (+5V) ---- 1

PROGRAM:

```
ASSUME CS:CODE,DS:DATA,ES:EXTRADATA

SEGMENT

STG1 DB  "ELECTRONICS"

DATA ENDS

EXTRA SEGMENT

STG2 DB 11 DUP(?)

EXTRA ENDS CODE

SEGMENT START:

        MOV AX,DATA

        MOV DS,AX MOV

        AX,EXTRAMOV

        ES,AX CLD

        LEA SI,STG1

        LEA DI,STG2

        MOV CX,11

        REP MOVSB

        INT 03H

CODE ENDS

END START
```

OPCODE:

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

B) INSERT A STRING

PROGRAM

```
ASSUME CS:CODE,DS:DATA,ES:DATADATA

SEGMENT

S1 DB "ANURAGLEGE"S2

DB "COL"

S3 DB 0DH DUP(?)

DATA  ENDS CODE

SEGMENT START:

        MOV AX,DATA

        MOV DS,AX

        MOV ES,AX STD

        LEA SI,[S1+09H]

        LEA DI,[S3+0CH]

        MOV CX,04H REP

        MOVSB LEA

        SI,[S2+2]

        MOV CX,3

        REP MOVSB
        LEA SI,[S1+5]

       MOV CX,6

        REP MOVSB

       INT 03H

CODE ENDS

END START
```

OPCODE:

MEMORYLOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

C)REVERSING A STRING

PROGRAM

```
 ASSUME CS: CODE, DS: DATA ,ES: DATA

 DATA SEGMENT

 S1 DB "KNOWLEDGE"S2

 DB 09H DUP (?) DATA

 ENDS

 CODE SEGMENT

 START:

        MOV AX,DATA

        MOV DS,AX

        MOV ES,AX LEA

        SI,[S1] LEA

        DI,[S2+8] MOV

        CX,9

BACK:CLD

        LODSB
        STD STOSB

        DEC CX

        JNZ BACK

        INT 03H

 CODE ENDS
 END START
```

OPCODE:

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

D)DELETE CHARACTER OF STRING PROGRAM :

```
ASSUME CS:CODE,DS:DATA,ES:DATA
        DATA SEGMENT
        S1 DB "UNIVERSITY"S2
        DB 07H DUP (?) DATA
        ENDS
        CODE SEGMENT
        START:
                MOV AX,DATA
                MOV DS,AX
                MOV ES,AX CLD
                LEA SI,S1 LEA
                DI,S2 MOV
                CX,04HREP
                MOVSB
                LEA SI,[S1+7]
                MOV CX,03H
                REP MOVSB
                INT 03H
        CODE ENDS
        END START
```

OPCODE:

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

C) STRING LENGTH

PROGRAM

```
ASSUME CS:CODE,ES:DATA

DATA SEGMENT

STG DB "ANURAG#"

CHAR DB "#"

LEN DW 00H DATA

ENDS CODE

SEGMENTSTART:

        MOV AX,DATA

        MOV ES,AX CLD

        LEA DI,STG MOV

        AL,CHARMOV

        CX,14

        REPNE SCASB

        MOV LEN,DI

INT 03H

CODE ENDS

END START
```

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

D) STRING COMPARISION

CASE: STRINGS ARE EQUAL

```
ASSUME CS:CODE, DS: DATA, ES: DATADATA
SEGMENT
S1  DB  "ELECTRONICS  $" S2
DB  "ELECTRONICS  $" S3  DB
"COMPUTER $"
MSG1 DB "STRINGS ARE EQUAL $" MSG2 DB
"STRINGS ARE NOT EQUAL $"DATA ENDS
CODE SEGMENT
START:
        MOV AX,DATA
        MOV DS,AX
        MOV ES,AX LEA
        SI,S1
        LEA DI,S2
        MOV CX,11
        CLD
        REP CMPSB
        JNZ ABC
        LEA DX,MSG1
LAST:MOV AH,09H
        INT  21H MOV
        AH,4CHINT 21H
        ABC:LEA DX,MSG2
        JMP LAST
```

```
CODE ENDS
END START
```

CASE: STRINGS ARE NOT EQUAL

```
;COMPARE STRINGS
ASSUME CS: CODE, DS:DATA,ES:DATA DATA
SEGMENT
S1 DB "ELECTRONICS $" S2
DB "ELECTRONICS $" S3 DB
"COMPUTER $"
MSG1 DB "STRINGS ARE EQUAL $" MSG2 DB
"STRINGS ARE NOT EQUAL $"
DATA ENDS
CODE SEGMENT
START:
        MOV AX,DATA
        MOV DS,AX
        MOV ES,AX LEA
        SI,S1
        LEA DI,S3
        MOV CX,11
        CLD
        REP CMPSB
        JNZ ABC
        LEA DX,MSG1
LAST:MOV AH,09H
        INT 21H
```

```
            MOV AH,4CH
            INT 21H
            ABC:LEA DX,MSG2
            JMP LAST
    CODE ENDS
    END START
```

OPCODE:

MEMORY LOCATION	**OP-CODE**	**LABEL**	**INSTRUCTION**

INPUT:

OUTPUT:

RESULT:

Viva Questions:

1) What are the string manipulation instructions?

2) What are the repeat instructions?

3) What is the use of DUP instruction?

4) What is the meaning of ORG assembler Directive?

5)What is the relation between 8086 processor frequency & crystal Frequency

EXP.NO.:5 **DATE:**

INTERFACING ADC AND DAC TO 8086.

(A)PROGRAM FOR ANALOG TO DIGITAL CONVERTOR

AIM:
1. To write a program for conversion of analog data to digital output.
2. To write a program for conversion of digital data to analog output. The analog output will be in the form of triangular wave, saw tooth wave, square wave/rectangular wave.

APPARATUS:
1. 8086 Trainer.
2. Power supply for trainer and interface module.
3. A/D, D/A interface module.
4. Power mate connector.
5. FRC connector.
6. Cathode ray oscilloscope.

PROGRAM:

```
;8 bit ADC 0809 (SUCCESIVE APPROX. METHOD), 100 microsec.

;convertion time is used to convert voltage signal

;simulated by on board pot. It provided varying voltage

;0 to 5v,connected to channel no.1. Processor interface

;is provided via 26 pin FRC.

            ADC SEGMENT

            ASSUME CS:ADC

               CR55 EQU 8807H

               PORTB EQU 8803H

               PORTC EQU 8805H

                        ORG 100H

                          START:

                                         MOV AX,0000H

                                          MOV ES,AX
```

```
          MOV SS,AX
          MOV AX,11F0H ;Init .SP
          MOV SP,AX
          PUSH CS         ;Set CS=DS
          POP DS
          MOV DX,CR55 ;Init port A,B MOV
          AL,81H            ;C(upper) as OP
          OUT DX,AL       ;C(lower) as IP
          MOV DX,PORTB
          MOV AL,00H
          OUT DX,AL
          MOV DX,CR55
          MOV AL,09H        ;Set PC4(ALE) bit
          OUT DX,AL       ;high
          MOV AL,08H        ;Set PC4 bit to
          OUT DX,AL       ;latch
          MOV AL,83H        ;Set portB as IP OUT
          DX,AL           ;rest same as before
          INT 0ACH
COVN: MOV DX,CR55  ;Set PC6(start ofMOV
          AL,0DH            ;convertion) OUT
          DX,AL
          MOV AL,0CH
```

```
                OUT DX, AL MOV
                DX, PORTC
         BACK: IN AL, DX          ;Check PC1(EOC) low
                AND AL, 02H        ;to insure convertion
                JNZ BACK
     COVNCHK: IN AL, DX           ;Convertion really
                AND AL, 02H        ;Completed
                JZ COVNCHK ;Yes, then set
                MOV AL, 0BH         ; PC5(OE) to read
                MOV DX, CR55
                OUT DX, AL
                MOV DX, PORTB ;Read digital dataIN
                AL, DX
                MOV CL, AL
                MOV DX, CR55
                MOV AL, 0AH
                OUT DX, AL INT
                0ABH MOV AL,
                02H MOV DX, CX
                NOP
                MOV DH,00H
                INT 0AEH
```

```
                  MOV AH,0BH
                  INT 0A1H AND
                  AL,0FFH
                  JZ COVN      ;Start next sample INT
                  0A3H              ;Return to monitor
         ADC ENDS
           END START
```

OPCODE:

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

(B)PROGRAM FOR DIGITAL TO ANALOG CONVERTOR

PROGRAM:

```
ASSUME CS: CODE
CODE SEGMENT
START:
        MOV DX, 8006H
        MOV AL, 80H OUT
        DX.AL MOV AL,
        00H MOV DX,
        8000H
   UP: OUT DX, AL
        INC AL
        JMP UP
        RET
CODE ENDS
END START
```

OPCODE:

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

RESULT:

Viva Questions:

1) Which is by default pointer for CS/ES?

2) How many segments present in it?

3) What is the size of each segment?

4) Basic difference between 8085 and 8086?

5)Which operations are not available in 8085?

EXP.NO.:6 **DATE:**

PARALLEL COMMUNICATION BETWEEN TWOMICROPROCESSORS USING 8255.

AIM: To write an alp for parallel communication between two microprocessors by using 8255.

APPARATUS: 8086 Trainer kit-2, 8255, Power Supply and connectors.

PROCEDURE:

1) Connect 8255 card (Periware – 3) to your kit through 50 pin FRC.
2) Connect port A tags to GND, and Vcc through patch cords,
3) Make S1 towards you to enable Single Stepping.
4) Connect PC4 tag with patch cord to Vcc.
5) Connect PC3 tag to RST 5.5 tag through patch cord.
6) L18 (on MB) position should be 2-3 short & L3 on converter card position should be 1-2.
7) Enter the software code as given in list 3(A).
8) Execute the program by pressing G, CR, CR, SR_SEGM 145E, CR, ADDR 0100, CR & observe

PROGRAM:

```
0103  B8 10FF       MOV  AX,10FFH      ;Init of SP for Kit
 0106  8B E0        MOV  SP,AX         ;On PC
 0108 0E            PUSH CS            ;Init for DS
 0109 58            POP  AX            ;Load kit INTS
 010A  8E D8        MOV  DS,AX         ;(A0-BF)
 010C 90            NOP                ;In PC using
 010D 90            NOP                ;CALL in place
 010E 90            NOP                ;of 3 NOP`S
 010F  B8 0000      MOV AX,0000H       ;Load ES as 0000H
 0112  8E C0        MOV ES,AX
 0114  B0 90        MOV  AL,90H        ; Control Word for Mode 0, ;A-input
0116 BA 01E6        MOV DX,01E6H       ; CSW Address in DX
 0119  EE           OUT DX,AL          ; Write csw (OUT 90 to 01E6H)
 011A  BA 01E0      MOV DX,01E0H       ; Port A Address in DX
 011D  EC           IN  AL,DX          ; Read port A (     IN from 01E0H)
 011E                     ISR055:
 011E  26: A2 1000        MOV ES:[1000H],AL        ; Store received data in memory.
 0122  CD AC              INT 0ACH                 ;Clear display
 0124  BB 0140 R          MOV BX,OFFSET MSG        ;LXI  H,00H  43H, Pointer for
                                                   ;look up table.
 0127  CD AF              INT 0AFH                 ;CALL OUTMSG, Calls display
                                                   ;routine.
 0129  B3 01              MOV BL,01H               ; Set counter for delay.(C)
 012B  B9 00FF            LOOP:  MOV CX,00FFH   ;LXI  D,FFH  FFH, Wait to stabilise
                                                   the display.
 012E  CD AA             INT 0AAH                  ;Call delay, Calls delay routine.

 0130  26: A0 1000       MOV AL,ES:[1000H]         ;Retrive port A received data for
                                                   ;display purpose.
 0134  8A D0             MOV DL,AL
 0136  B0 02             MOV AL,02H                ; Number of digits to be
                                                   ; displayed in data segment.
 0138  CD AE             INT 0AEH                  ;CALL  NMOUT,  NM  OUT  routine  is
013A  FE CB              DEC BL
 013C  75 ED             JNZ  LOOP
 013E  CD A4             INT 0A4H                  ;GOTO Command Mode.
 0140  50 4F 54 41 20 20   MSG:  DB "POTA  ",03H    ; End of text.  03
 0147                     X86P55A1 ENDS
                                END  STRT
```

OUTPUT:

	Data Bus	$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A0	A1	Comments
Start	90	L	-	L	1	1	Control Word Mode 0, port A - input.
Step1	DATA	L	L	-	-	-	Read data from port A.
Step2	-	-	-	-	-	-	Shows input data on kit display .

RESULT:

Viva Questions:

1.Expand USART?

2.Where do we prefer the serial communication?

3.What is the function of instruction pointer (IP) register?

4. What is the difference between IN and OUT instructions

5. What is MODEM

EXP.NO.:7 **DATE:**

INTERFACING TO 8086 AND PROGRAMMING TOCONTROL STEPPER MOTOR

AIM: Write an Assembly Language Program to rotate the Stepper Motor in clockwise as wellas anti-clockwise direction.

APPARATUS: 8086 Trainer kit, Stepper,
Motor Interface Card,
Stepper Motor,
Power supply.

PROGRAM:(STEPPER CLOCK WISE)

```
ASSUME CS:CODE

CODE SEGMENT

START:     MOV DX,8006H

            MOV AL,80H

            OUT DX,AL

            MOV DX,8000H

            MOV AL,88H

            BACK:OUT DX,AL

            CALL DELAY

            ROR AL,01H

            JMP BACK

            DELAY:MOV CX,0FFFH

            L1:DEC CX

            JNZ L1

            RET

CODE ENDS

 END START
```

PROGRAM: (ANTICLOCK WISE)

```
 ASSUME CS: CODE
 CODE SEGMENT
 START:
 MOV DX,8006H
 MOV AL,80H
 OUT DX,AL
MOV DX,8000H
MOV AL,88H
 BACK:OUT DX,AL
 CALL  DELAY
ROL AL,01H
 JMP BACK DELAY:MOV CX,0FFFH
L1:DEC CX
JNZ L1
RET
 CODE ENDS
 END START
```

OPCODE:

MEMORY LOCATION	OP-CODE	LABEL	INSTRUCTION

INPUT:

OUTPUT:

RESULT:

Viva Questions:

1) What is the difference between min mode and max mode of 8086?

2) What is the difference between near and far procedure?

3) What is the difference between Macro and procedure?

4) What is the difference between instructions RET & IRET?

5) What is the difference between instructions MUL & IMUL?

EXP.NO.: 8(a) **DATE:**

ARITHMETIC OPERATIONS USING 8051

MULTIBYTE ADDITION

AIM: To write an Assembly Language Program to perform Multibyte addition using 8051.

APPARATUS:

- 8051 Microcontroller kit
- Keyboard
- Power supply

ALGORITHM:

1 Start the program.

2 Assign the address 4200 to Data pointer & load the contents.

3 Move the content 00h into R3 register.

4 Move the contents of external data memory into A register.

5 Move the content of A register into R0.

6 Increment the content of data pointer.

7 Move the contents of external data memory into A register.

8 Perform addition operation with the content of A register to R0 content and result is stored in A register.

9 Move the content of A register to R1 register.

10 Clear the content of A register.

11 Increment the content of data pointer.

12 Move the contents of external data memory into A register.

13 Move the contents of A register into R0 register.

14 Increment the content of data pointer.

15 Move the contents of external data memory into A register.

16 Perform addition operation with carry the content of A register to R0 content and result is stored in A register.

17 Jump if no carry to label loop

18 Increment R3 register.

19 Move the contents of A into R2 register.

20 Move the contents of R1 into A register.

21 Increment the content of data pointer.

22 Move the contents of A register into external data memory.

23 Move the contents of R2 into A register.

24 Increment the content of data pointer.

25 Move the contents of A register into external data memory

26 Move the contents of R3 into A register.

27 Increment the content of data pointer.

28 Move the contents of A register into external data memory.

29 Stop the program.

PROGRAM:

ADDRESS	OPCODE	LABEL	MNEMONICS	OPERANDS	COMMENTS
			MOV	DPTR, #4200	
			MOV	R3,#00	
			MOVX	A,@DPTR	
			MOV	R0,A	
			INC	DPTR	
			MOVX	A,@DPTR	
			ADD	A,R0	
			MOV	R1,A	
			CLR	A	
			INC	DPTR	
			MOVX	A,@DPTR	
			MOV	R0,A	
			INC	DPTR	
			MOVX	A,@DPTR	
			ADDC	A,R0	
			JNC	LOOP	
			INC	R3	
		LOOP	MOV	R2,A	
			MOV	A,R1	
			INC	DPTR	
			MOVX	@DPTR,A	
			MOV	A,R2	
			INC	DPTR	
			MOVX	@DPTR,A	
			MOV	A,R3	
			INC	DPTR	
			MOVX	@DPTR,A	
		HLT	SJMP	HLT	

RESULT:

INPUT:

OUTPUT:

EXP NO:8(b) **Date:**

ARITHMETIC OPERATIONS USING 8051
MULTIBYTE SUBTRACTION

AIM: To write an Assembly Language Program to perform Multibyte subtraction using 8051.

APPARATUS:

- 8051 Microcontroller kit
- Keyboard
- Power supply

ALGORITHM:

1 Start the program.
2 Assign the address 4200 to Data pointer & load the contents.
3 Move the content 00h into R3 register.
4 Move the contents of external data memory into A register.
5 Move the content of A register into R0.
6 Increment the content of data pointer.
7 Move the contents of external data memory into A register.
8 Perform subtraction operation with borrow the content of A register to R0 content and result is stored in A register.
9 Move the content of A register to R1 register.
10 Clear the content of A register.
11 Increment the content of data pointer.
12 Move the contents of external data memory into A register.
13 Move the contents of A register into R0 register.
14 Increment the content of data pointer.
15 Move the contents of external data memory into A register.
16 Perform subtraction operation with borrow the content of A register to R0 content and result is stored in A register.
17 Jump if no borrow to label loop
18 Increment R3 register.
19 Move the contents of A into R2 register.
20 Move the contents of R1 into A register.
21 Increment the content of data pointer.

22. Move the contents of A register into external data memory.
23. Move the contents of R2 into A register.
24. Increment the content of data pointer.
25. Move the contents of A register into external data memory
26. Move the contents of R3 into A register.
27. Increment the content of data pointer.
28. Move the contents of A register into external data memory.
29. Stop the program.

PROGRAM:

ADDRESS	OPCODE	LABEL	MNEMONICS	OPERANDS	COMMENTS
			MOV	DPTR, #4200	
			MOV	R3,#00	
			MOVX	A,@DPTR	
			MOV	R0,A	
			INC	DPTR	
			MOVX	A,@DPTR	
			SUBB	A,R0	
			MOV	R1,A	
			CLR	A	
			INC	DPTR	
			MOVX	A,@DPTR	
			MOV	R0,A	
			INC	DPTR	
			MOVX	A,@DPTR	
			SUBB	A,R0	
			JNC	LOOP	
			INC	R3	
		LOOP	MOV	R2,A	
			MOV	A,R1	
			INC	DPTR	
			MOVX	@DPTR,A	
			MOV	A,R2	
			INC	DPTR	
			MOVX	@DPTR,A	
			MOV	A,R3	
			INC	DPTR	
			MOVX	@DPTR,A	
		HLT	SJMP	HLT	

RESULT:

INPUT:

OUTPUT:

EXP NO:8(c) **Date:**

ARITHMETIC OPERATIONS USING 8051
MULTIBYTE MULTIPLICATION

AIM: To write an Assembly Language Program to perform Multibyte multiplication using 8051.

APPARATUS:

- 8051 Microcontroller kit
- Keyboard
- Power supply

ALGORITHM:

1 Start the program.

2 Assign the address 4000 to Data pointer & load the contents.

3 Move the contents of external data memory into A register.

4 Move the content of A register into B register.

5 Increment the content of data pointer.

6 Move the contents of external data memory into A register.

7 Perform multiplication operation with the content of A register with B register.

8 Assign the address 4200 to destination Data pointer.

9 Move the contents of A register into external data memory.

10 Increment the content of data pointer.

11 Move the contents of B into A register.

12 Move the contents of A register into external data memory.

13 Stop the program.

PROGRAM:

ADDRESS	OPCODE	LABEL	MNEMONICS	OPERANDS	COMMENTS
			MOV	DPTR, #4000	
			MOVX	A,@DPTR	
			MOV	0F0,A	
			INC	DPTR	
			MOVX	A,@DPTR	
			MUL	AB	
			MOV	DPTR,#4200	
			INC	DPTR	
			MOV	A,0F0	
			MOVX	@DPTR,A	
		HLT	SJMP	HLT	

RESULT:

INPUT:

OUTPUT:

EXP NO: 8(d) **Date:**

ARITHMETIC OPERATIONS USING 8051
MULTIBYTE DIVISION

AIM: To write an Assembly Language Program to perform Multibyte division using 8051.

APPARATUS:

- 8051 Microcontroller kit
- Keyboard
- Power supply

ALGORITHM:

1 Start the program.
2 Assign the address 4000 to Data pointer & load the contents.
3 Move the contents of external data memory into A register.
4 Move the content of A register into B register.
5 Increment the content of data pointer.
6 Move the contents of external data memory into A register.
7 Perform division operation with the content of A register by B content.
8 Assign the address 4200 to destination Data pointer.
9 Move the contents of A register into external data memory.
10 Increment the content of data pointer.
11 Move the contents of B into A register.
12 Move the contents of A register into external data memory.
13 Stop the program.

PROGRAM:

ADDRESS	OPCODE	LABEL	MNEMONICS	OPERANDS	COMMENTS
			MOV	DPTR, #4000	
			MOVX	A,@DPTR	
			MOV	0F0,A	
			INC	DPTR	
			MOVX	A,@DPTR	
			DIV	AB	
			MOV	DPTR,#4200	
			INC	DPTR	
			MOV	A,0F0	
			MOVX	@DPTR,A	
		HLT	SJMP	HLT	

RESULT:

INPUT:

OUTPUT:

VIVA QUESTIONS:

1. Define Micro Controller.
2. What is the difference b/w Microprocessor& Micro controller?
3. How Physical address is generated?
4. What is the function of 01h of Int 21h?
5. Which are pointers present in this 8086?

EXP NO: 9 **Date:**

TIMER IN DIFFERENT MODES

AIM: To write an Assembly Language Program to perform timer in different modes.

APPARATUS:

- 8051 Microcontroller kit
- Keyboard
- Power supply
- RS 26 core cable
- CRO
- Probes

ALGORITHM:

1 Start the program.

2 Note the keyboard value register using time zero.

3 FFF2h is load on N to TH0, TL0.

4 PL.3 toggled for high & low pulses.

5 Delay sub ordering using the time is called.

6 In subording time 0 is started by the set B TR0 instructions.

7 Timer 0 counts the passing of the each clock which is provided by crystal oscillator as the timer counter with goes to the states of FF & FF3 ,FF4, FF5 &…and till reaches FFFFH are more clock is rows it zero raising the time zero TR0=1 at that point JNB instruction.

8 Timer 0 is stopped by the instruction clear the TR0, delay subordinating hence & process is repeated.

9 Stop the program.

PROGRAM:

```
        MOV     TMOD,#01
HERE : MOV      TL0,#0F2H
        MOV      TH0,#0FFH
        CPL      P1.5
        ACALL  DELAY
        SJMP     HERE
DELAY:SET B     TR0
```

```
                JNB     TF0,AGAIN
        AGAIN:  CLR     TR0
                CLR     TE0
                RET
```

INPUT:

OUTPUT:

RESULT:

1. What is the reset address of 8086?

2. What is the size of flag register in 8086?

3.Explain all. 3. What is the difference between 08H and 01H functions of INT 21H?

4. Which is faster- Reading word size data whose starting address is at even or at odd address of memory in 8086?

5. Which is the default segment base: offset pairs?

EXP.NO.:10 **DATE:**

PROGRAM AND VERIFY INTERRUPT HANDLING IN8051.

1. AIM: To write an Assembly Language Program to generate an interrupt using 8259 Programmable Interrupt Controller with 8086 Microprocessor.

2. APPARATUS:

- ESA 86E trainer kit
- 8259 Programmable Interrupt Controller kit
- Personal Computer

3. PROGRAM:

```
ORG 2000H;          Set PC value to 2000H
MOV AX, 0000H;      Initialize Segment Registers
MOV CS, AX
MOV ES, AX
MOV SS, AX
MOV SP, 3000H;      Initialize Stack Pointer
                    ; Interrupt Vector Initialization
MOV SI, 0120H;      INT 0 Vector address 0120H is the base of Interrupt Vector Table.
MOV AX, 2200H
MOV [SI], AX
ADD SI, 02H
MOV AX, 0000H
MOV [SI], AX
ADD SI, 02H;        Interrupt 1 Vector Address
MOV AX, 2210H
MOV [SI], AX
ADD SI, 02H
MOV AX, 0000H
MOV [SI], AX
ADD SI, 02H;        Interrupt 2 Vector Address
MOV AX, 2220H
MOV [SI], AX
ADD SI, 02H
MOV AX, 0000H
MOV [SI], AX
ADD SI, 02H;        Interrupt 3 Vector Address
MOV AX, 2230H
MOV [SI], AX
ADD SI, 02H
MOV AX, 0000H
MOV [SI], AX
ADD SI, 02H;        Interrupt 4 Vector Address
MOV AX, 2240H
MOV [SI], AX
ADD SI, 02H
MOV AX, 0000H
MOV [SI], AX
ADD SI, 02H;        Interrupt 5 Vector Address
MOV AX, 2250H
MOV [SI], AX
ADD SI, 02H
MOV AX, 0000H
MOV [SI], AX
ADD SI, 02H;        Interrupt 6 Vector Address
MOV AX, 2260H
MOV [SI], AX
ADD SI, 02H
MOV AX, 0000H
MOV [SI], AX
ADD SI, 02H;        Interrupt 7 Vector Address
MOV AX, 220H
MOV [SI], AX
```

```
ADD SI, 02H
MOV AX, 0000H
MOV [SI], AX         ; 8259 INTIALIZATION
MOV DX, 0FFC8H;       Indicates Port address of 8259
MOV AL, 17H;          ICW1 initialization (IC4 needed, Single, Interval 4, edge
                     triggered)
OUT DX, AL
MOV DX, 0FFCAH;       ICW2 (Multiple for int vector address table) for masking 120H
                     as base address of Interrupt Vector Table)
MOV AL, 48H
OUT DX, AL
MOV AL, 03H;         ICW4 (8086 mode, auto EOI)
OUT DX, AL
MOV AL, 00H
OUT DX, AL
STI;                 (Set Interrupt Flag) Enable INTR of 8086 trainer kit
HERE: JUMP HERE
ORG 2100H; MESSAGES FOR ISRs
MSG0: DB 20H, 20H, 0AH,'INT0 OCCURRED', 0AH, 0DH
MSG1: DB 20H, 20H, 0AH,'INT1 OCCURRED', 0AH, 0DH
MSG2: DB 20H, 20H, 0AH,'INT2 OCCURRED', 0AH, 0DH
MSG3: DB 20H, 20H, 0AH,'INT3 OCCURRED', 0AH, 0DH
MSG4: DB 20H, 20H, 0AH,'INT4 OCCURRED', 0AH, 0DH
MSG5: DB 20H, 20H, 0AH,'INT5 OCCURRED', 0AH, 0DH
MSG6: DB 20H, 20H, 0AH,'INT6 OCCURRED', 0AH, 0DH
MSG7: DB 20H, 20H, 0AH,'INT7 OCCURRED', 0AH, 0DH
ORG 2200H; INT0 ISR
CLI
LEA DX, MSG0
JMP DISP
INT 03H
```

```
ORG 2210H; INT1 ISR
CLI
LEA DX, MSG1
JMP DISP
INT 03H
ORG 2220H; INT2 ISR
CLI
LEA DX, MSG2
JMP DISP
INT 03H
ORG 2230H; INT3 ISR
CLI
LEA DX, MSG3
JMP DISP
INT 03H
ORG 2240H; INT4 ISR
CLI
LEA DX, MSG4
JMP DISP
INT 03H
ORG 2250H; INT5 ISR
CLI
LEA DX, MSG5
JMP DISP
INT 03H
ORG 2260H; INT6 ISR
CLI
LEA DX, MSG6
JMP DISP
INT 03H
ORG 2270H; INT7 ISR
```

```
            CLI
            LEA DX, MSG7
            JMP DISP
            INT 03H
            ORG 2300H; COMMON DISPLAY ROUTINE FOR ALL ISRs
    DISP:    MOV SI, DX
            MOV CX, 11H
    L1:      MOV AL, [SI]
            CALL FAR 0FE00:0000H; CALL ROUTINE TO DISPLAY THE MSGS
            INC SI
            LOOP L1
            STI
            IRET
```

4. PROCEDURE:

1. Open win 86E window and initialize PC address as 20000H
2. Enter the instruction until entire program is completed and click on the disassembly icon.
3. Provide the connections between 8086 and 8259 as follows

JP1=23

JP2=23

JP3=23

JP4=23

JP5=23

JP6=23

JP7=23

JP8=23

JP9=12

JP10=12

4. Go to command prompt and give G 2000H.
5. Specific interrupt can be selected by 4 ways DIP switch selection for different interrupts are as follows:

3	2	1	4 Ways
0	0	0	IR0
0	0	1	IR1
0	1	0	IR2
0	1	1	IR3
1	0	0	IR4

1	0	1	IR5
1	1	0	IR6
1	1	1	IR7

6. While Program is running press the PUSH button on the 8259 kit to provide service for specific interrupt service routine.

INPUT:

OUTPUT:

RESULT:

Viva Question:

1)How many no. of interrupts available for 8051?

2) Which is the highest priority interrupt for 8051?

3) What is an ISR and IVT?

4)What is the difference between software and hardware interrupt?

5) What is the vector address for serial communication interrupt?

EXP. NO.: 11 **DATE:**

UART OPERATION IN 8051

AIM: Write ALP Of UART operation in 8051.

APPARATUS:

1. 8051 trainer kit with keyboard
2. Talk with PC
3. RPS
4. RS – 232
5. FRC cables
6. UART Module

PROGRAM FOR MODE-0-TRANSMITTER:-

```
Org 9000h
MOV SCON,#00H (SCOON=98)
UP1:MOV R7,#8H
MOV A,#80H(SBUF=99)
Up:CLRTi(Ti=99)
MOV SBUF,A
XX:JNBTi,XX
CLR P1.0
 SETB P1.0
LCALL DELAYRR A
DJNZ R7,UP
JMP UP1
Delay: MOV R0,#0FFH
Up3:MOV R1,#0FFH
Up2:DJNZ R1,UP2
DJNZ R0,UP3
RET
```

PROGRAM FOR MODE-0-RECIEVER:

```
ORG 9000h
MOV SCON,#11H (SCOON=98)
Up1:CLR P1.(P1.0=90)
```

```
 CLR P3.(P3.1=B1)
SETB ri(Ri=99)
SETB P1.0
CLR Ri(SBUF=99)
XX: JNB Ri,XX
MOV A,SBUF
MOV R6,A
LCALL DELAY
SJMP UP
```

RESULT:

VIVA QUESTION:

1.What is macros?

2.What is TEST instruction?

3.What is LEA instruction?

4.What are status keys in keyboard?

5.What operands we can declare?

EXP NO: 12 **Date:**

INTERFACING MATRIX OR KEYBOARD TO 8051

AIM: Interface a Keyboard to 8051 microcontroller.

APPARATUS:

1.8051 trainer kit with keyboard

2.Key board module

3. RPS

4.FRC cables

5. RS-232 cable

PROGRAM:

```
CNTRL        EQU   2043H                  ;CONTROL PORT ADDRESS OF  8255
PORTA        EQU   2040H         ;PORTA ADDRESS OF 8255
PORTB        EQU   2041H         ;PORTB ADDRESS OF 8255
PORTC        EQU   2042H         ;PORTC ADDRESS OF 8255
```

```
Org 9000h
MOV A,#90H
MOV DPTR,#CNTRL
MOVX @DPTR,A
MOV B,#20H
Blink 2: MOV DPTR,#PORTB
MOV A,#FFH
MOVX @DPTR,A
MOVDPTR,#PORTC
MOV A,#00H
```

```
MOVX @DPTR,A
MOVA,#F0H
 MOVX @DPTR,AD
JNZ B,BLNK2
Back: MOV A,#FEH
MOV B,#21H
Blink1: MOV DPTR,#PORTB
MOVX @DPTR,A
MOV DPTR,#PORTC
 MOVA,#00H
MOVX@DPTR,A
MOVA,#F0H
MOVX @DPTR,AL
CALL DELAY
 RL A
DJNZ B BLNK1
SJMP BACK
Delay:   MOV R0,#F7H O
loop:  MOVR1,#FFH I
loop: DJNZ R1,ILOOP
DJNZ R0,OLOOP
RET
```

INPUT:

OUTPUT:

RESULT:

Viva Question:

1. What is the size of flag register?
2. Can you perform 32 bit operation with 8086? How?
3. Whether 8086 is compatible with Pentium processor?
4. What is 8087? How it is different from 8086?
5. While accepting no. from user why u need to subtract 30 from that?

ADVANCED EXPERIMENTS

EXP NO: 1 **Date:**

INTERFACING WITH 8086

8255 PROGRAMMABLE PERIPHERAL INTERFACE

AIM: To write an Assembly Language Program to interfacing peripheral interface with 8086 microprocessor such that port A and port B of 8255 will acts as output ports.

APPARATUS:

- ESA 86E trainer kit
- Power supply
- 8255 study pad
- keyboard
- Serial data bus

ALGORITHM:

1 Start the program.

2 Initialize 8255 as output.

3 Move input data to AL.

4 Output the data at port A.

5 Invert the input data.

6 Insert and output the data at port B.

7 Introduce delay and repeat.

8 Stop the program.

PROGRAM:

ADDRESS	OPCODE	INSTRUCTION	COMMENT
		MOV DX,0FFC6	Initialize 8255 point as output
		MOV AL,80	
		OUT DX,AL	
		MOV AL,90	
		MOV AL,55	
		MOV DX,0FFC0	Output data at port A
		NOT AL	
		MOV DX,0FFC2	Insert data and output the value of port B
		OUT DX,AL	
		JMP 2006	Introduce delay & repeat

RESULT:

INPUT:

OUTPUT:

VIVA QUESTIONS:

1. Define Interrupt.

2.What is mean by PPI?

3.Define DMA.

4.What are the Software Interrupts?

5. What is mean by UART?

EXP NO:2 **Date:**

INTERFACING WITH 8051

8279 KEYBOARD INTERFACE

1.AIM: To write an Assembly Language Program to display string ESA in the display field of the study card using 8279 keyboard and display controller decode method with 8051 microcontroller.

2. APPARATUS:

- 8051 Microcontroller
- Power supply
- 8279 Keyboard and Display controller

3. ALGORITHM:

1 Start the program.

2 Initialize the starting address .

3 Divide the clock frequency.

4 Initialize 8279 interfacing unit.

5 Enter the data with right entry instruction perform the decode operation for data.

6 Scan the keyboard.

7 Introduce the reading table to the 8051to read the value from I/O devices.

8 Initialize the input data at 9000 location.

9 Stop the program.

4. PROGRAM:

ADDRESS	OPCODE	LABEL	INSTRUCTION	COMMENT
8000			ORG 8000H	
8000	90F181		MOV DPTR,#F181	
8003	74 FF		MOV A,#FF	
8005	F0		MOVX @DPTR,A	
8006	7A 90		MOV R2,#90	
8008	7B 00		MOV R3,#00	
800A	90F181		MOV DPTR,#F181	
800D	74 90		MOV A,#90	
800F	F0		MOVX @DPTR,A	
8010	74 11		MOV A,#11H	
8012	F0		MOVX @DPTR,A	

8013	78 08		MOV R0,#08H	
8015	74 00		MOV A,#00H	
8017	90F180	STAR	MOV DPTR,#F180	
801A	F0	T:	MOVX @DPTR,A	
801B	18		DEC R0	
801C	B800F6		CJNE R0,#0,8015	
801F	79 00		MOV R1,#00	
8021	909000		MOV DPTR,#9000	
8024	E4		CLR A	
8025	93		MOVC A,@A+DPTR	
8026	90F180		MOV DPTR,#F180	
8029	F0		MOVX @DPTR,A	
802A	09		INC R1	
802B	909090		MOV DPTR,#9000	
802E	E9		MOV A,R1	
802F	B904F3	STAR	JNE R1,#04,8025	
8032	80FE	T1:	SJMP 8032	
9000			ORG 9000H	
9000	0497D6		DB 04H,97H,D6H	
9003	770404		DB 77H,04H,04H	
9006	00		DB 00H	
		TABL		
		E:		

5. RESULT:

INPUT:

OUTPUT:

Viva Questions:

1. What is Digital Clock?

2. What are the applications of Digital Clock?

3. What is the formula for frequency?

4. Why clock is required?

5. What pins are used in 8085 to connect the clock?